



SandS: Social & Smart

Social housekeeping through intercommunicating appliances
and shared recipes merged in a pervasive web-services
infrastructure

Delivery number D8.6

User Manual

Project	
Project acronym:	SOCIAL&SMART
Project Full title:	Social housekeeping through intercommunicating appliances and shared recipes merged in a pervasive web-services infrastructure
Grant agreement no.:	317947
Document	
Deliverable number	D8.6
Deliverable title:	User Manual
Work Package:	WP8
Due date of deliverable:	30.7.2014
Lead beneficiary:	NTUA
Editors:	G. Siolas, G. Caridakis, G. Stratogiannis, A. Lipaj, I. Mlakar, B. Apolloni
Contributing beneficiaries:	AMIS, UNIMI
Reviewer:	B. Apolloni
Status:	Final Version
Version and date	V0.6 21/07/2015
Changes	Final revision

Project co-funded under the SEVENTH FRAMEWORK PROGRAMME THEME 3 ICT -
INFORMATION AND COMMUNICATIONS TECHNOLOGIES

Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	CO Confidential, only for members of the Consortium (including the Commission Services)	

Project co-ordinator: Prof. Bruno Apolloni
Università degli studi di Milano
Tel: +39 02 50316284
E-mail: apolloni@di.unimi.it
Project website address: www.sands-project.eu

Executive Summary

Current deliverable corresponds to the User manual of the SandS system from the end user's perspective. The SandS UI stands at the top layer of the SandS architecture and communicates with the other layers via the DI. The UI framework is based on SOA architecture, designed as a web-based application with underlying infrastructure developed in PHP, JavaScript and HTML/CSS3 technologies. The UI aims to enable the eahouker to interact with different entities and functionalities in the SandS ecosystem such as recipes, appliances, sensors and scheduled tasks. The deliverable is structured in two main sections: 2 Instantiation and 3 Runtime. The former focuses on the initial registration and setup of the user's profile and domestic settings while the latter on actual functionalities of the SandS system. The typical workflow that was adopted is user and appliances registration followed by the typical request, recipe, and evaluation triplet SandS approach. More advanced features of the interface in both the setup stage, such as long term scheduling and home rules, as well as the runtime stage have been also included in the respective sections.

Table of contents

1	User Interface overview.....	4
2	Instantiation	10
2.1	Registration/Login	10
2.2	Home screen	12
2.3	Editing user profile	14
2.4	Appliances registration	15
2.5	Home rules	19
2.6	Scheduling.....	21
2.7	Sensors.....	24
3	Runtime.....	25
3.1	Task request	25
3.2	Recipe approval/modification	28
3.3	Messaging	32
3.4	Evaluation	33

1 User Interface overview

The SandS UI framework (SandS front-end) represents the top most layer of the SandS framework. It is designed for development and delivery of SandS user interfaces. The UI framework is based on SOA architecture, designed as a web-based application with underlying infrastructure developed in PHP, JavaScript and HTML/CSS3 technologies. It is capable to serve several eahouker engagement strategies, e.g. send user requests to lower layer of SandS framework, as well as receive requests, in form of messages, from lower layers of SandS framework and overlay them to eahouker's. General functional outline of SandS UI framework is given in Figure 1. As outlined in Figure 1, the developed UI framework cover the following engagement strategies:

- a) *Engagement with recipes*, architecture and interfaces for delivery of eahouker's engagement with the life cycle of a recipe. The principal function of the Sands is to achieve user goals (tasks) via recipes sent to DI as a response to specific task descriptions as provided through the use of natural language.
- b) *Engagement with appliances*, architecture and interfaces for delivery of eahouker's engagement with appliances located and assigned to his/hers domestic environment. Through the developed interfaces the eahouker can ascertain what appliances are assigned to his/her domestic environment and, engage/disengage additional appliances.
- c) *Engagement with domestic environment*, architecture and interfaces for adapting the functionality of eahouker's domestic environment via engagement rules (home rules). These rules correlate operational aspects of appliances with basic parameters of domestic environment such as: noise, temperature, humidity, power/water consumption, time of the day and human presence.
- d) *Engagement with ancillary service*, architecture and interfaces for adjusting information regarding eahouker's profile/location and, for making use of the messaging service which provides a reverse link between eahouker and DI.

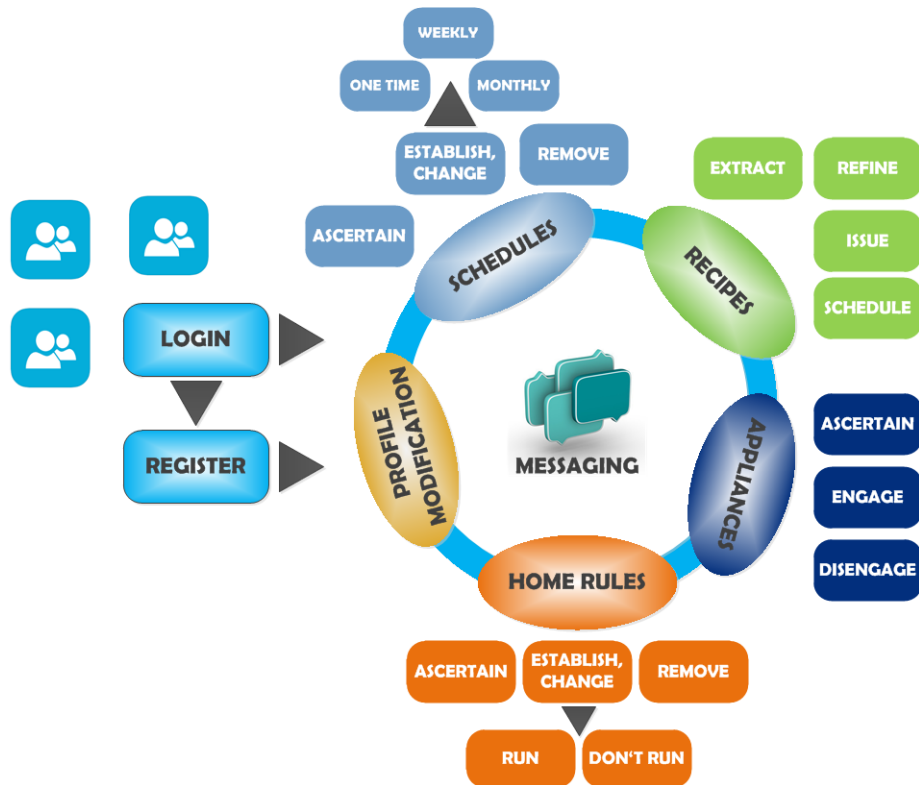


Figure 1 Functional outline of SandS UI framework

The main entry point of the SandS UI framework is the “*LOGIN*” page. Through this page existing eahoukers may login and new eahoukers may register. The “*LOGIN*” page requires from the eahouker to insert his/hers credentials (username, password) and, as a security mechanism, solves a simple mathematical expression. The “*REGISTER*” page requires the eahouker to provide his/hers personal information such as first name, surname, date of birth, gender, social status, and number of children. Furthermore the eahouker has to provide information regarding his credentials, e.g., desired username, password and e-mail address. If no username is given, the system generates its automatically based on eahoukers personal information. Finally, the eahouker must also provide certain level of detail regarding his/hers domestic environment. The required level of detail involves parameters such as: type of location, address of location including street, street number, city and country and, the type of residence. After the form has been correctly populated and submitted, the eahouker is registered and automatically redirected towards his/hers home screen.

From the home screen the user may access one of the five engagement strategies. The messaging service initiates automatically. Its results (messages) are placed within the UI’s interfaces as to be always present and accessible regardless of the engagement strategy the eahouker is involved in. The home screen was designed in line with the outline provided in Figure 1. As shown in Figure 2, it offers four “icons”:

- the “My profile icon”, through which eahouker may engage ancillary service for modifying personal information.
- “Recipes icon”, through which eahouker may enroll in engagement with recipes strategy from the perspective of recipes extraction, refinement and issuing/scheduling
- “Appliances icon”, through which eahouker may enroll in engagement with appliances.
- “Planning & Configuration icon”, through which eahouker may enroll in engagement domestic environment from the perspective of schedules and home rules.

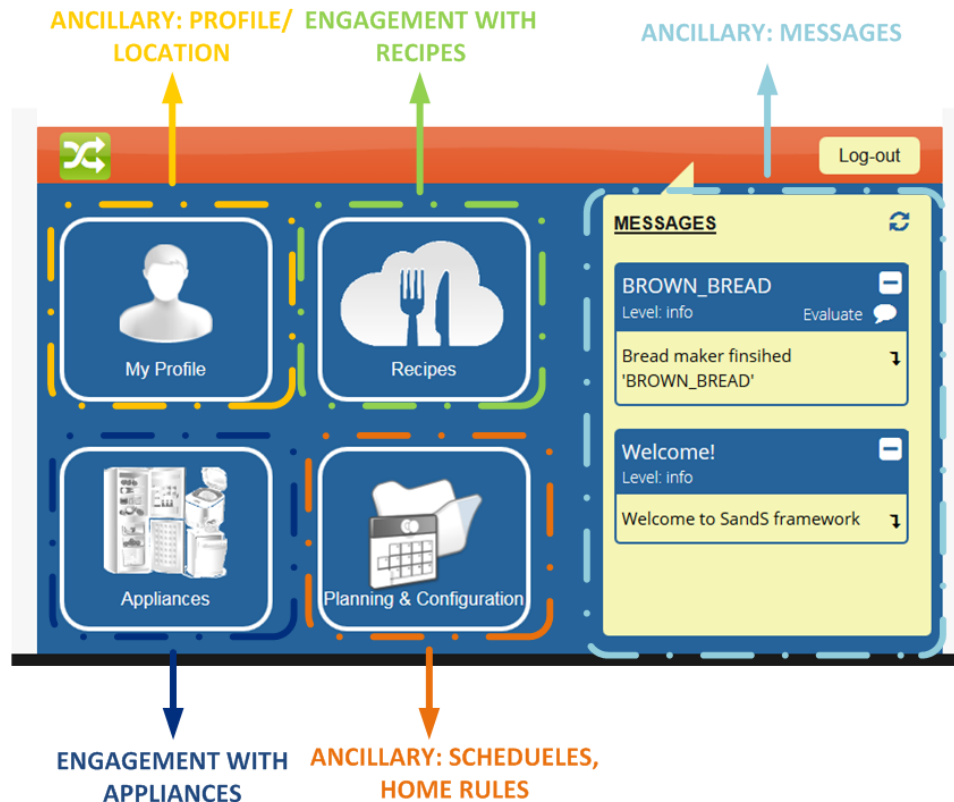


Figure 2 Home screen of SandS UI framework

The “My profile icon” overlays to the eahouker the interfaces through which he/she may change and update his/hers personal information such as: first name, surname, date of birth, gender, social status, and number of children. Furthermore the eahouker may also update information regarding his/hers domestic environment such as: type of location, address of location including street, street number, city and country and, the type of residence. When the eahouker submits new information the content of the form is first verified against several rules, such as:

- 1 None of the fields may be empty or unselected
- 2 Email format validated by using the following regular expression: $\wedge[A-Z0-9._\%+-]+\@[A-Z0-9.-]+\.[A-Z]{2,4}\$/i$

If the form is not verified the eahouker is prompted to correct the data. If however the correctness of the content in the form is verified, the information is passed to the DI and updated accordingly.

The “Recipes icon” overlays to the eahouker the interfaces through which he/she may interact with recipes engagement strategy of SandS framework. The developed interfaces and underlying web services allow for a user to query for a recipe by describing the task in natural language. A natural language processor (NLP) tries to identify the targeted appliance as well as appliance-dependent recipe’s parameters based on the task description provided as text-input (or through speech in Chrome web-browser). When the extraction process completes the appliance-dependent form used for requesting a recipe is overlaid and automatically populated with extracted parameters. While on this form user may further adjust the parameters of the task. Using these parameters as well as eahouker’s social information (e.g. similarity with other eahoukers, eahouker’s history, etc.) a most suitable recipe is selected/generated by lower layers of SandS framework. The recipe is returned to FE and overlaid to the eahouker. On the ‘recipe’s detail page’ the eahouker may totally agree with the suggested recipe. As such he can either issue the recipe for immediate execution (e.g. as soon as possible) or schedule it as a one time or periodic task, executed on daily, weekly or monthly basis. If however the eahouker

does not agree (or agrees partially) with the recipe, he/she may also refine (readjust) the parameters based on which the recipe was chosen/generated. An appropriate, appliance-dependent form enables him/her to alter preference parameters in order to avoid receiving the same recipes. It also provides the eahouker with input masks for fixing some parameters as a priori constraints for the next recipe (e.g. fixing the ingredients, fixing the total execution time or even fixing values at instructions level, such as changing temperature, motor spin or even the duration of instruction sequence). Upon receiving the new recipe the eahouker may again issue, schedule or refine the recipe.

After a recipe is completed a message is sent to the eahouker asking him/her to evaluate the executed recipe. Eahouker evaluation is tightly connected to his/her expectation about the task s/he requested. In turn, it is critical for the learning algorithms at the basis of recipe generation. For this purpose the eahouker satisfaction is evaluated via dynamically generated appliance-dependent evaluation forms. The forms may be accessed via messages triggered upon the completion of a recipe (e.g. when DI terminates the recipe). The forms are designed as simple questionnaire imposing questions directed to collect the user-experience regarding the aspects best describing the results of the recipe. For each of these aspects the eahouker selects a value from the 'scale' awful to 'perfect'.

The "Appliances icon" overlays to the eahouker the interfaces through which he/she may interact with appliances engagement strategy of SandS framework. The developed interfaces and underlying web services allow for a user to ascertain what appliances are connected to his/hers domestic environment. The appliances are, based on their type, grouped into six categories: *bread makers*, *washing machines*, *dishwashers*, *coffee pots*, *refrigerators* and *ovens*. By selecting a category, the details of engaged appliances within the category (if any) are overlaid. Through the category details users may engage additional appliances of the selected type or disengage previously engaged appliances. The eahouker may also update priori parameters of appliances (e.g. MAC address and password).

The "Planning & Configuration icon" overlays to the eahouker the interfaces through which he/she may interact with ancillary services regarding scheduling and home rules. Long-term scheduling serves to plan the execution of tasks through recipes. Long term scheduling is available for the eahoukers through a front-end interfaces corresponding to cron-like commands. The long-term planning is intended for scheduling different tasks in advance. Each schedule may be instanced via recipes engagement strategy. The GUIs developed within the schedule context of the SandS UI framework allow for user to preview already instanced schedules, remove them or redefine them. The GUIs developed for redefinition of schedules enable for the eahouker to manage and specify both periodic recipe execution (e.g. perform certain tasks on daily/weekly/monthly basis), and singular recipe execution (one time task) planned in advance (e.g. perform certain task only once, but rather on a certain day at a certain hour, rather than immediately). By selecting the "redefine" option the current parameters of the schedule are loaded and populated on a form. Through this form any aspect of a schedule, with exception to the recipe and appliance type, may be completely redefined.

Home rules are used in order to correlate tasks/recipes with the basic parameters of domestic environment, such as: noise, temperature, humidity, power/water consumption, time of the day and human presence. By using home rules the eahouker may design the operational context of the domestic environment to his/hers own needs. For instance, the user may specify a set of criteria/conditions which will prevent an appliance to operate and execute planned or immediate tasks. Furthermore, the eahouker may also create sets of conditions which, when met, trigger the execution of a task (recipe) on the selected appliance. Through the developed interfaces the eahouker can ascertain what home-rules he/she were already established in the targeted domestic environment and also add /remove additional home rules. The list of rules may also be filtered by appliance type, type of action or validation variable (the parameter of domestic environment).

The general idea behind the creation of home rules is that eahouker first selects a parameter (the validation variable) which will preside over the criteria/condition. Next the eahouker selects the

relation (inequality) of the validation variable and its values (e.g. less than, equal, greater than, between). Afterwards the eahouker selects/inserts the appropriate value(s). By doing this he/she creates criteria. In the next step the user must select the type of action which SandS framework must perform if and when the criteria are met. The “don’t run” option implies that the eahouker wishes to prevent an appliance to execute tasks. If it is selected, the user must select over which appliances in his/hers domestic environment the newly created criteria will preside. The ‘run option’ implies that eahouker wishes to execute a task on an appliance when the criteria are met. The recipe and the appliance are selected via UI’s and web services designed for the recipes engagement strategy of SandS framework. Rather than issuing/scheduling the recipe eahouker selects it as a part of the home rule. Finally when the eahouker confirms the rule is submitted to the DI and saved as part of eahoukers home rule collection.

The messaging strategy of SandS framework was developed in order to deliver a bidirectional communication channels through which lower SandS layer may initiate the interaction and request for users engagement. The messaging functionality is implemented through a PHP/jquery based implementation of WebSockets (RFC 6455). The WebSocket Protocol enables two-way communication between a client running untrusted code in a controlled environment to a remote host that has opted-in to communications from that code. The security model used for this is the origin based security model commonly used by web browsers. The developed user interfaces allow the eahouker to receive and monitor any messages generated by lower layers of SandS framework, regardless the interface he/she is currently working on. The messaging system also supports message queuing which allows for any message that is yet not manually removed to be recollected and redisplayed in the interface. For the purpose of messaging a segment of the UI implements HTML-based message holder which is constant and persistent through all engagement strategies of SandS. In its current implementation the messaging system supports two contexts of messages, e.g. a general context named as ‘any’ and recipes context named as ‘evaluation’.

The general context is used to convey general messages regarding SandS framework and eahouker’s domestic interface. Such messages may be read and/or deleted. In order to distinguish between the general context of messages three levels are created:

- INFO – intended for general messages regarding operational and other contexts of SandS framework (e.g. conveying current temperature of a room, informing a new eahouker was registered to the interface, informing about actuation of a home rule, etc.)
- WARNING – intended for conveying warnings about some non-terminal condition which is not terminal, however eahouker’s intervention is desired. Namely the condition might cause a problem/error or failure in the future (e.g. a sensor has failed; oven light has failed, etc.).
- ERROR – intended for conveying unexpected conditions which indicate that a desired operation has failed. Error messages are used when user intervention is absolutely required (e.g. fridge door is open, oven is open and a recipe cannot be executed, the motor of bread maker has failed and is not responding, etc.).

The ‘evaluation’ context is used when SandS conveys to the user that a task has been completed and the system seeks for evaluation of the result to be submitted. When such a message is received it can be read and removed. Additionally it provides links through which the eahouker may access the appropriate evaluation form.

Finally, a part of SandS UI framework is also devoted to the implementation of privacy and security mechanism. Namely, web-based and IoT vulnerabilities now outnumber traditional computer security concerns and as measured by Google, about one in ten web pages may contain malicious code. In the years 2013 and 2014 cybercrime still remains prevalent and, still represents a damaging threat to both businesses and consumers. Internet privacy involves the right or mandate of personal privacy concerning the storing, repurposing, provision to third parties, and displaying of information

pertaining to oneself via the Internet. In order to ensure user privacy and anonymity of data the consortium of SandS project has devised a privacy policy which follows the W3C recommendations.

Furthermore, the data transmitted between Front-end and DI uses HTTPS/SSL and HTTPS/TLS levels of protocol security. Since front end is designed as general web-based solution providing services via php/ajax, HTML, JavaScript and CSS3 technologies appropriate security mechanism for the following major threats, identified for web pages and applications, were developed and delivered within the SandS UI framework: data injection (SQL injection) attack, cross-site scripting (XSS and XFS) techniques and ‘password-guessing attack’ known as a brute-force attack.

2 Instantiation

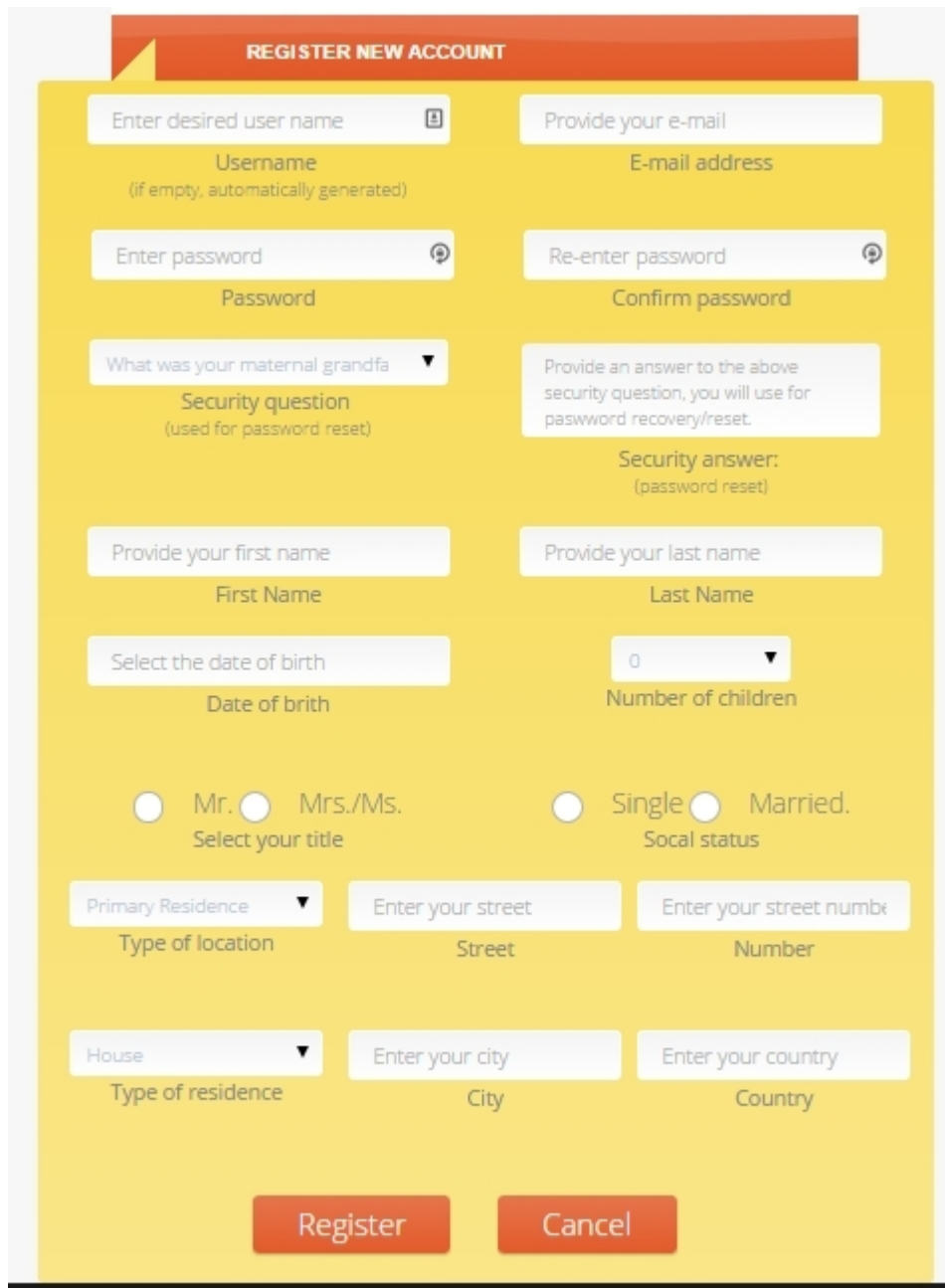
2.1 Registration/Login




The image shows a login screen with a yellow background and an orange header. The header contains the SandS logo. The main content area has the text "Please login!" at the top. Below this, there are three input fields: "User name:" with a placeholder "username", "Password:" with a placeholder "password", and a human verification question "Pleas enter the result of the equation" with the equation "8 - 7 =". Each input field has a small asterisk icon and a small number "2" next to it. At the bottom, there are two blue buttons: "Register as new" and "Log In".

Figure 3 Login screen with human verification and registration option


The Registration/Login screen presents the user with the options to login directly to SandS by entering his username/password and a bot security check or registering as a new user.





REGISTER NEW ACCOUNT

Enter desired user name  Username
(if empty, automatically generated)

Provide your e-mail E-mail address

Enter password  Password

Re-enter password  Confirm password


What was your maternal grandfa  Security question
(used for password reset)

Provide an answer to the above security question, you will use for password recovery/reset.
Security answer:
(password reset)


Provide your first name First Name

Provide your last name Last Name

Select the date of birth Date of birth


0  Number of children

☐ Mr. ☐ Mrs./Ms. ☐ Single ☐ Married.
Select your title Social status

Primary Residence  Type of location

Enter your street Street

Enter your street number Number

House  Type of residence

Enter your city City

Enter your country Country

Register **Cancel**

Figure 4 User registration form

In order to register the user must fill a form containing the usual account security fields (username, password, e-mail), first and last names, demographic information (date of birth, number of children), gender, marital status and detailed location fields.

2.2 Home screen

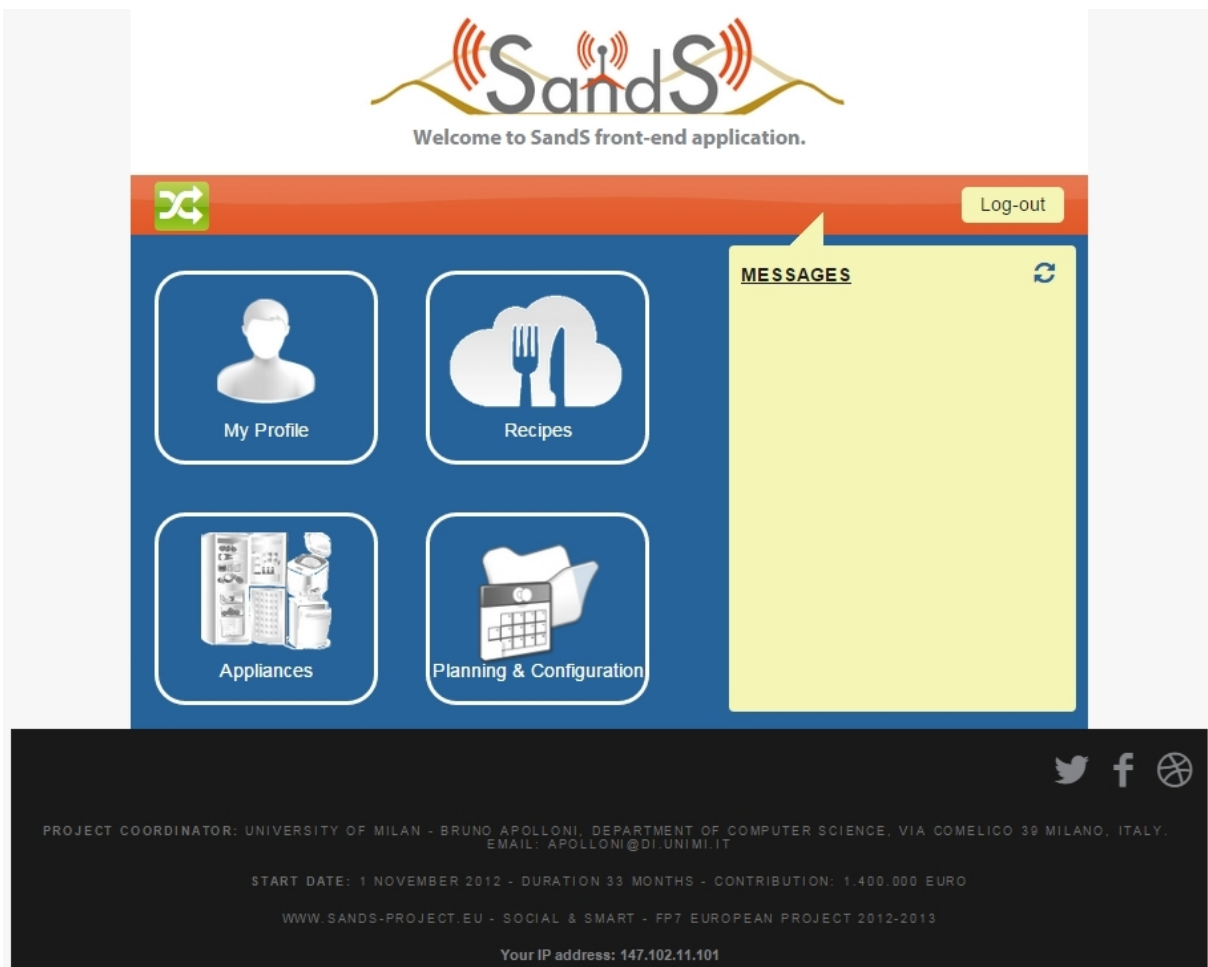


Figure 5 SandS home screen

After successfully login into the system, the user is greeted by SandS home screen. The home screen is made of four modules, “My Profile”, “Recipes”, “Appliances” and “Planning & Configuration”. By choosing “My Profile” the user can modify his profile (see Section 2.3 for details). By clicking “Recipes” the user can issue task requests, modify/refine recipes and approve their execution (Sections 3.1 and 3.2). “Appliances” serves to add and manage home appliances (Section 2.4) while “Planning & Configuration”, depicted in more detail in Figure 6, is for Long term scheduling (Section 2.6) and Home Rule (Section 2.5) and Senor (Section 2.7) management. At the right part of SandS home screen is situated the message board, where various messages are display for example related to recipe execution (Section 3.3) and also links for user feedback (Section 3.4).

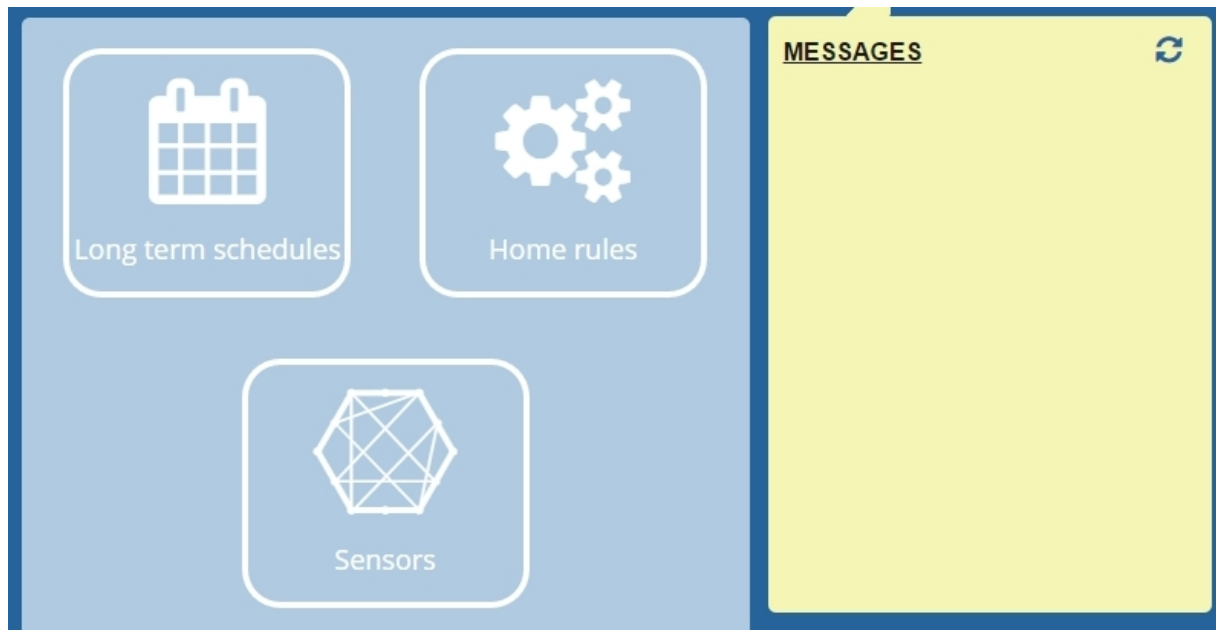
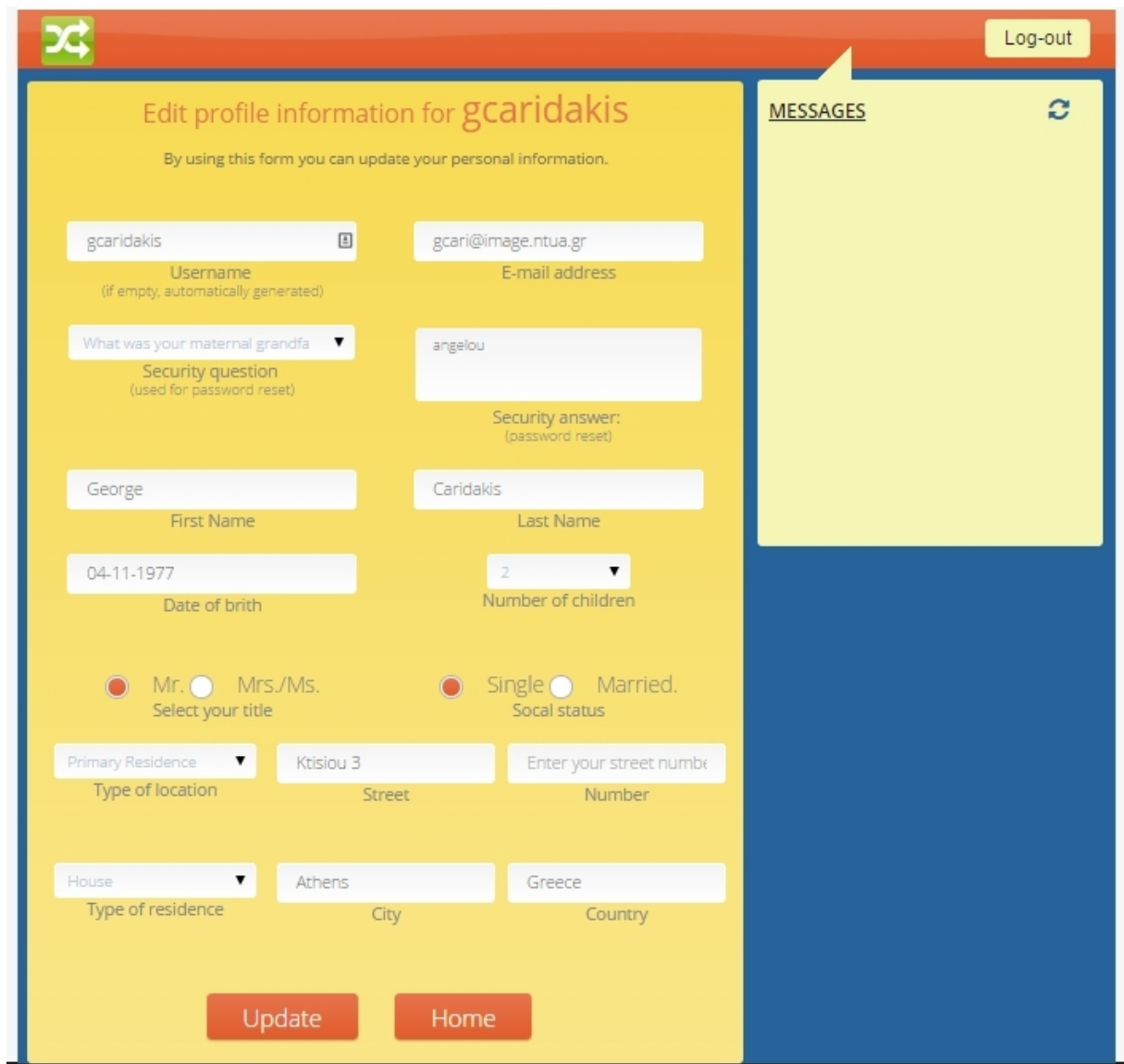


Figure 6 Planning and configuration control panel

2.3 Editing user profile



Edit profile information for gcaridakis

By using this form you can update your personal information.

Username: gcaridakis (if empty, automatically generated)

E-mail address: gcari@image.ntua.gr

Security question: What was your maternal grandfa (used for password reset)

Security answer: angelou (password reset)

First Name: George

Last Name: Caridakis

Date of birth: 04-11-1977

Number of children: 2

Select your title: ☒ Mr. ☐ Mrs./Ms.

Social status: ☒ Single ☐ Married.

Primary Residence: Ktisiou 3 (Type of location: Primary Residence)

Enter your street number

Type of residence: House (Type of residence)

City: Athens

Country: Greece

Update **Home**

Log-out

MESSAGES

Figure 7 User profile modification form

In “My profile” the user can modify all the fields that he has completed during registration (Figure 4).

2.4 Appliances registration

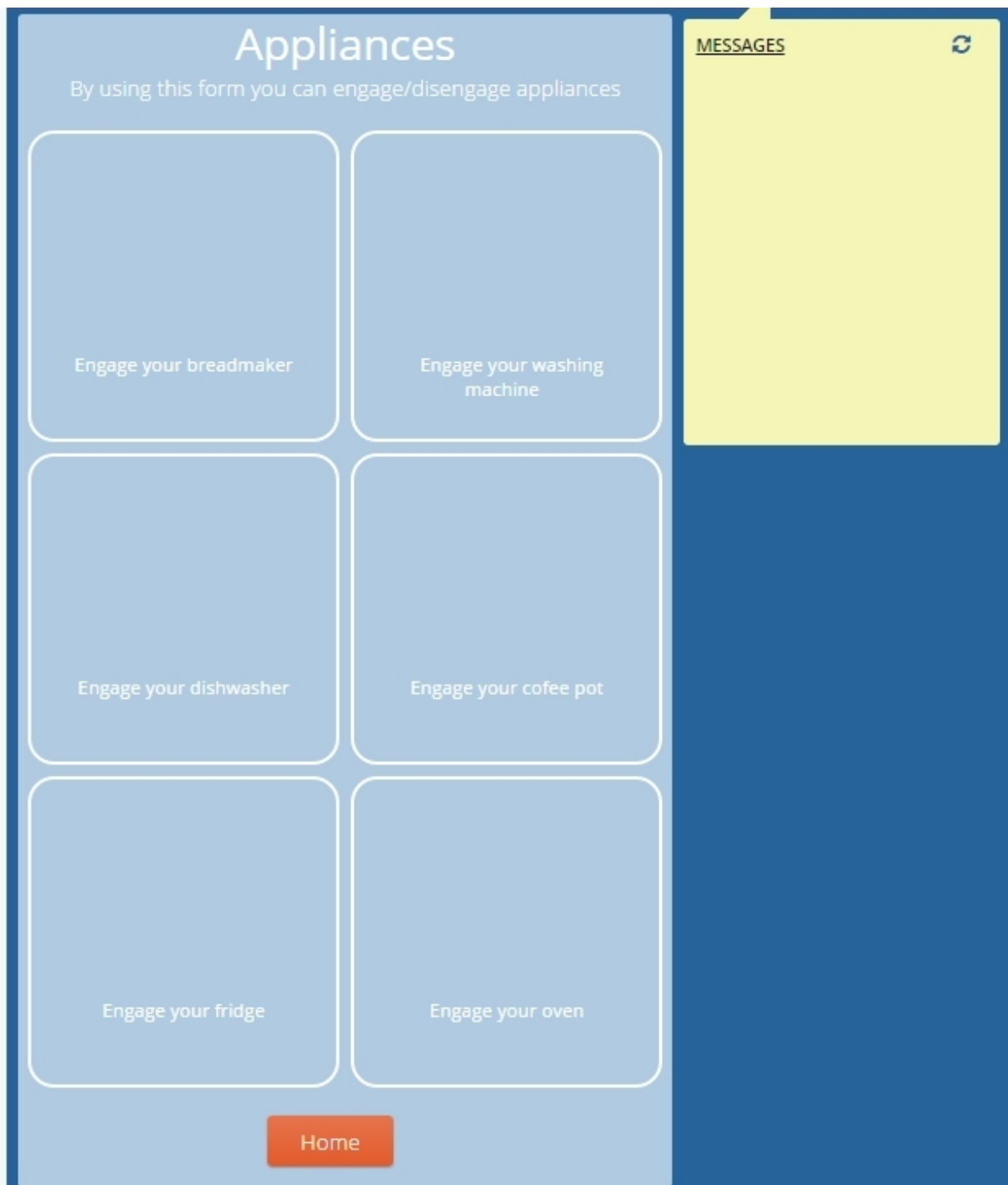
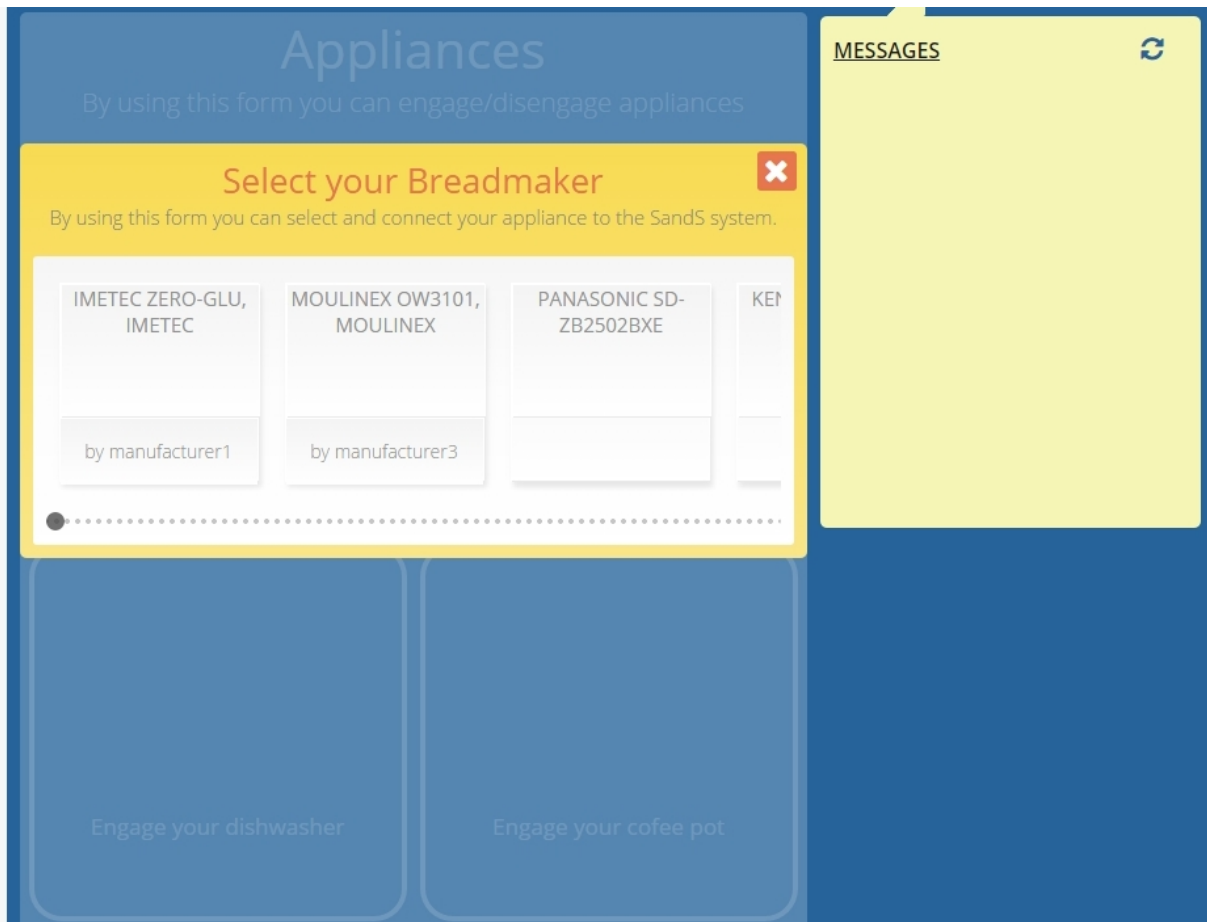


Figure 8 Appliance overview screen and type registration

The first step to add an appliance is to select the appliance type. Available appliance types consist of breadmaker, washing machine, dishwasher, coffee pot, fridge and oven.



The image shows a web interface for appliance registration. The main header is 'Appliances' with a subtitle 'By using this form you can engage/disengage appliances'. Below this is a yellow box titled 'Select your Breadmaker' with a subtitle 'By using this form you can select and connect your appliance to the SandS system.' Inside this box are three selection cards: 'IMETEC ZERO-GLU, IMETEC' (with 'by manufacturer1' below), 'MOULINEX OW3101, MOULINEX' (with 'by manufacturer3' below), and 'PANASONIC SD-ZB2502BXE'. To the right of these cards is a partially visible card for 'KEN'. Below the yellow box are two blue buttons: 'Engage your dishwasher' and 'Engage your cofee pot'. On the right side of the interface is a yellow 'MESSAGES' panel with a refresh icon.

Figure 9 Appliance registration model selection form

After having chosen the appliance type, he can select his specific appliance type from a list of available appliance models from various manufacturers.



Figure 10 Registered appliances

Figure 10 illustrates a user with two registered appliances, one Imetec Zero-GLU breadmaker and an Electrolux RWP1274TDW washing machine. The user has also three incoming messages in his message board.

Select your Breadmaker

By using this form you can select and connect your appliance to the SandS system.

CLATRONIC BBA
3365

GPDEPM2743

VZOU46Y3FS

Imetec Zero-GLU (Imetec)

manufacturer1

CE0682 31/12/2014

CERTIFICATE EXPIRES

CONSUMES

2_KWH POWER

TECHNICAL INFORMATION

0.920_KWH POWER

ENGAGE

Input mac-address

Insert MAC address

Input password

Insert password

Engage Cancel

Close form

Home

Figure 11 Registered appliances detailed information

As Figure 11 illustrates, the appliances detailed technical specifications available to the user.

2.5 Home rules

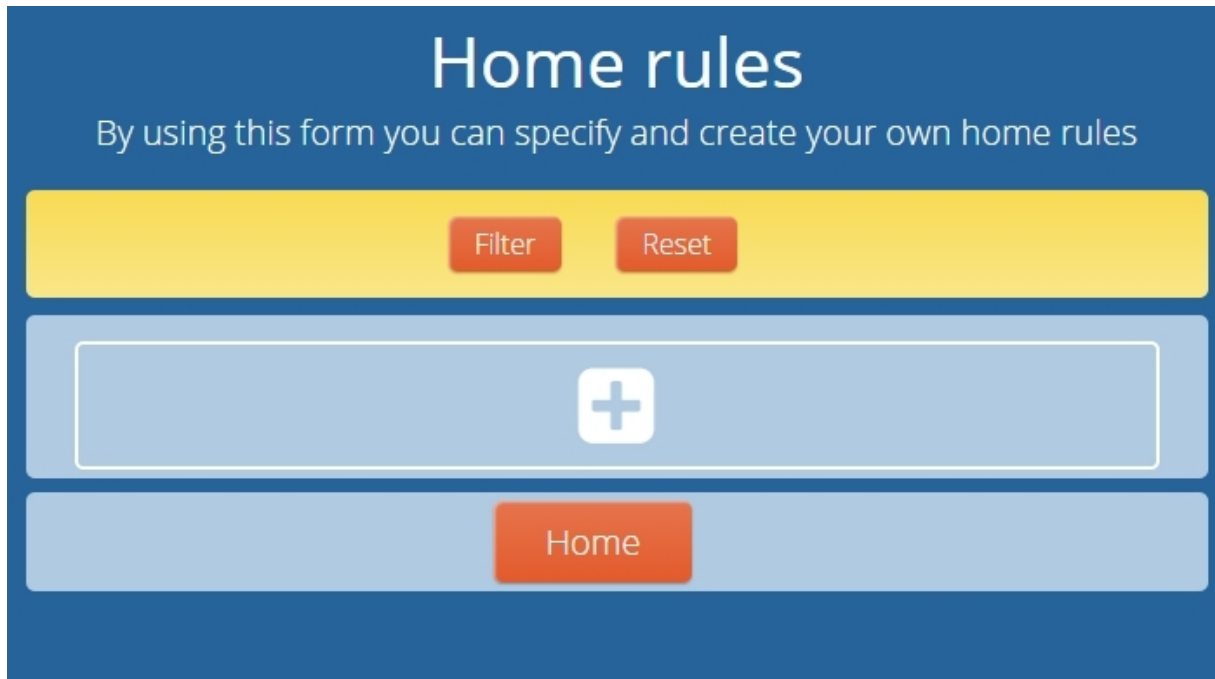
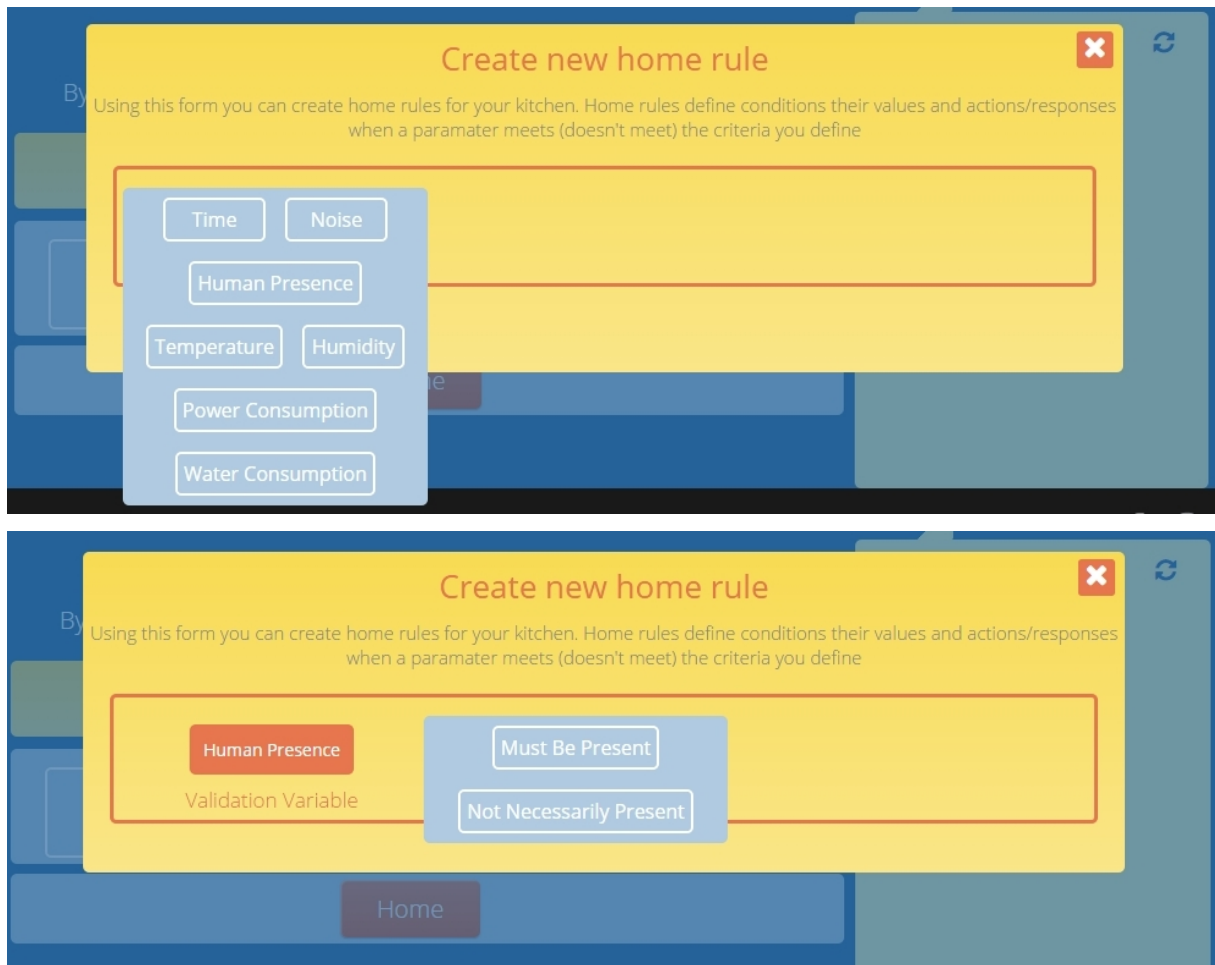


Figure 12 Home rules overview and addition form

Choosing “Home Rules” at the Planning and Configuration control panel (Figure 6) brings up a dialog where the user can add his own home rules.



Create new home rule

Using this form you can create home rules for your kitchen. Home rules define conditions their values and actions/responses when a paramater meets (doesn't meet) the criteria you define

Time Noise

Human Presence

Temperature Humidity

Power Consumption

Water Consumption

Human Presence

Must Be Present

Not Necessarily Present

Validation Variable

Home

Figure 13 Home rule condition selection form

To add a home rule, the user first selects a variable (Time, Noise, Human Presence, Temperature, Humidity, Power/Water Consumption) and then a condition (for example, “must be present” or the opposite).



Create new home rule

Using this form you can create home rules for your kitchen. Home rules define conditions their values and actions/responses when a paramater meets (doesn't meet) the criteria you define

Human Presence Yes

Validation Variable Must a user be present?

+ Add additional rule

Prevent Running Run A Recipe

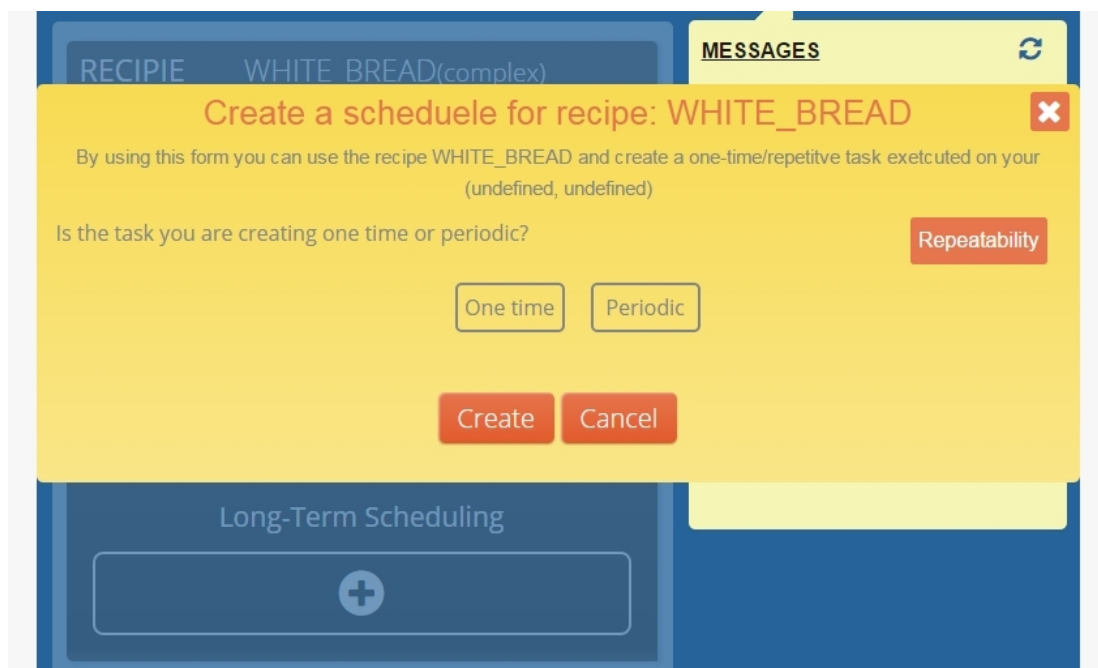
Please select an action appliance should/shouldn't do in regard to the above rules?

Please select appliances you dont wish to run

Figure 14 Home rule action selection form

Finally, the user defines the action to perform if the condition is met, for example run a recipe or prevent a recipe from running.

2.6 Scheduling



Create a schedule for recipe: WHITE_BREAD

By using this form you can use the recipe WHITE_BREAD and create a one-time/repitive task exetcutcd on your (undefined, undefined)

Is the task you are creating one time or periodic?

One time Periodic

Repeatability

Create Cancel

Long-Term Scheduling

+ Add additional rule

Figure 15 Recipe scheduling form

In order to schedule a recipe for execution, the user first must decide if its execution will be for one time or will be repeated periodically.

Create a schedule for recipe: WHITE_BREAD ✕

By using this form you can use the recipe WHITE_BREAD and create a one-time/repitive task exetcutud on your (undefined, undefined)

Is the task you are creating one time or periodic? Repeatability

How often would you like to perform the task? Repetitivity

At what days in the week do you wish to perform the task? Day of the week

(When you whish for the task to be performed several days a week, you can select one or more options.s.)

When do you wish to perform the task? Hour of the day

You can select several times. When generated each 'minutes' part will be added to each hour. (e.g. selecting 7:00 and 8:30 will result in CRON executed at 7:00, 7:30, 8:00 and 8:30)

Choose Time

Time 06:00

Hour

Cooking: 10 min 08 s at 61.461461

Figure 16 Recipe scheduling temporal options

Next the user can define the recipe execution day and time as well as the periodicity of the execution (Daily, Weekly and Monthly).

APPLIANCE	BRAND/MODEL	RECIPE	SCHEDULE
Breadmaker	Imetec,Imetec Zero-G LU	REPETIVITY PERIOD WHEN	Periodic Weekly Every Sun
Breadmaker	Imetec,Imetec Zero-G LU	REPETIVITY WHEN AT	One time 26-06-2015 17:00
Breadmaker	Imetec,Imetec Zero-G LU	REPETIVITY WHEN AT	One time 27-06-2015 05:00

[Home](#)

MESSAGES

DI Message
Level: info
Message

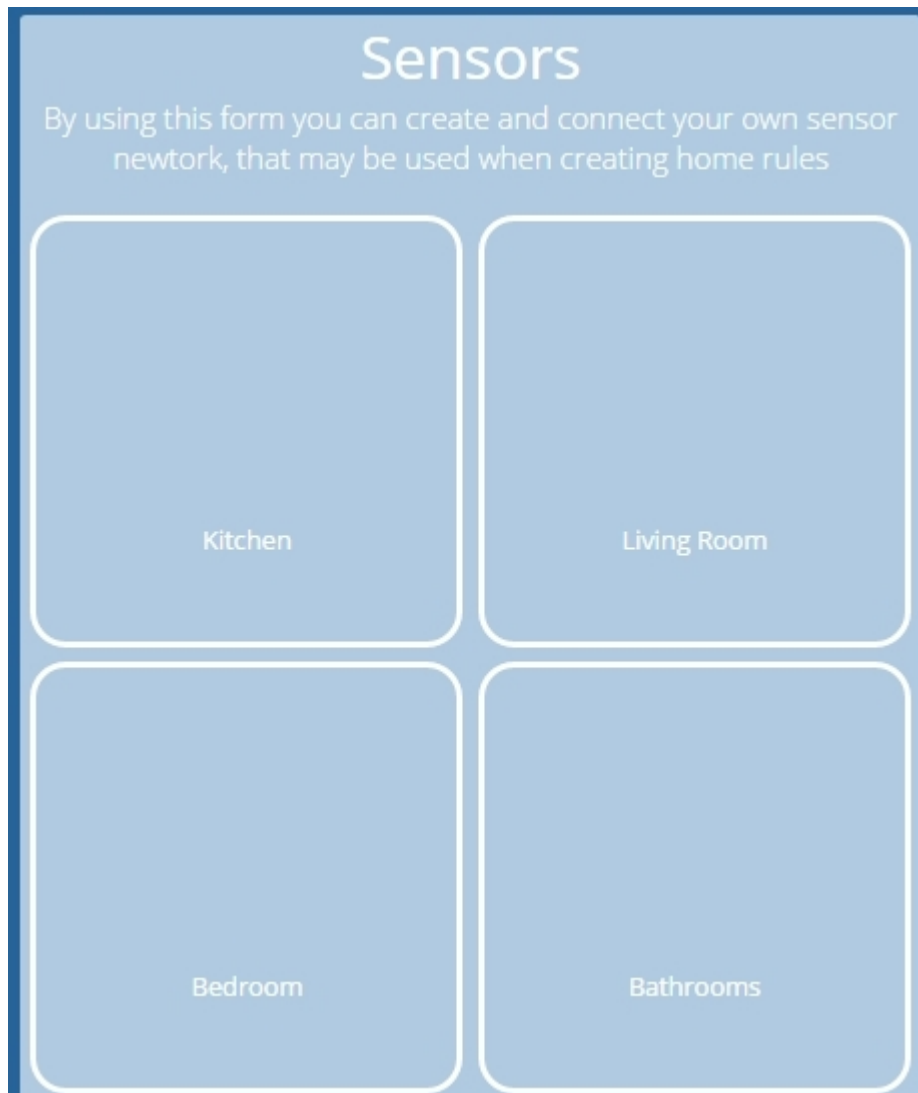
DI Message
Level: info
Message

DI Message
Level: info
Message

Figure 17 Scheduler overview

Figure 17 illustrates an example scheduler overview. Three breadmaker recipes are scheduled for execution, two one-time and one weekly periodic.

2.7 Sensors



The image shows a screenshot of a web application titled "Sensors". Below the title is a subtitle: "By using this form you can create and connect your own sensor network, that may be used when creating home rules". The main content area is divided into four large, light blue rounded rectangular boxes, each representing a room where a sensor can be added. The boxes are labeled "Kitchen", "Living Room", "Bedroom", and "Bathrooms".

Figure 18 Sensors overview and registration

Choosing “Sensors” at the Planning and Configuration control panel (Figure 6) brings up the Sensors window where the user can add ambient sensors situated in various rooms (kitchen, living room, bedroom, bathrooms). These sensors provide the real time context data needed to enforce the home rules of Figure 13.

3 Runtime

3.1 Task request

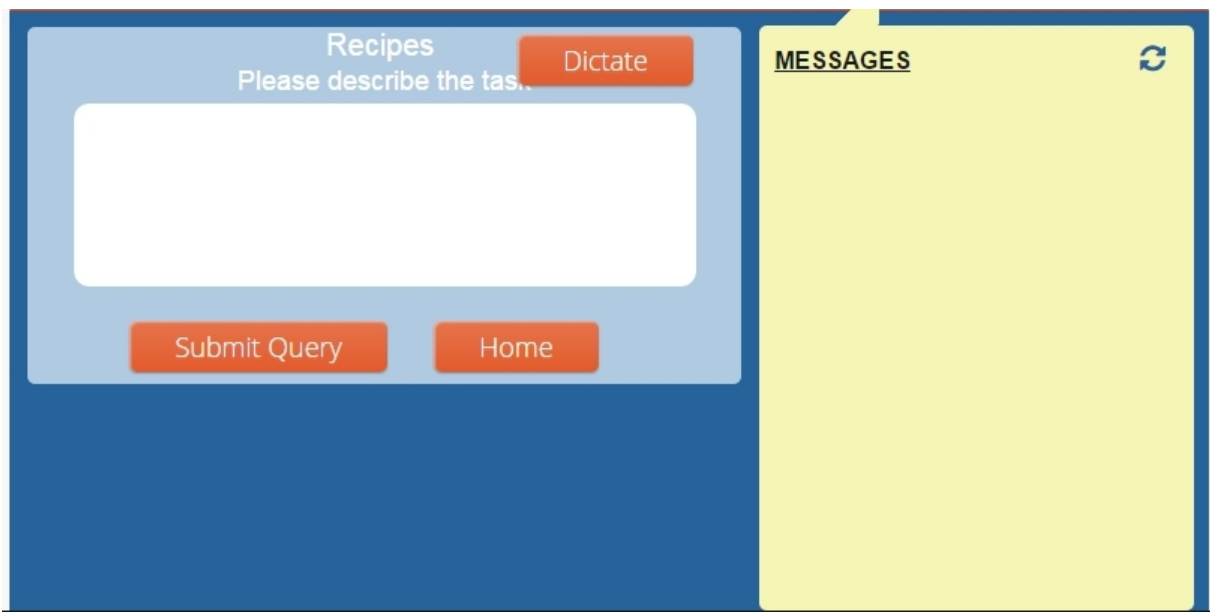


Figure 19 Task request home screen

Selecting “Recipe” from SandS home screen (Figure 5) brings up the Task request home screen. The user can type in natural language his task request e.g. “wash delicate clothes”

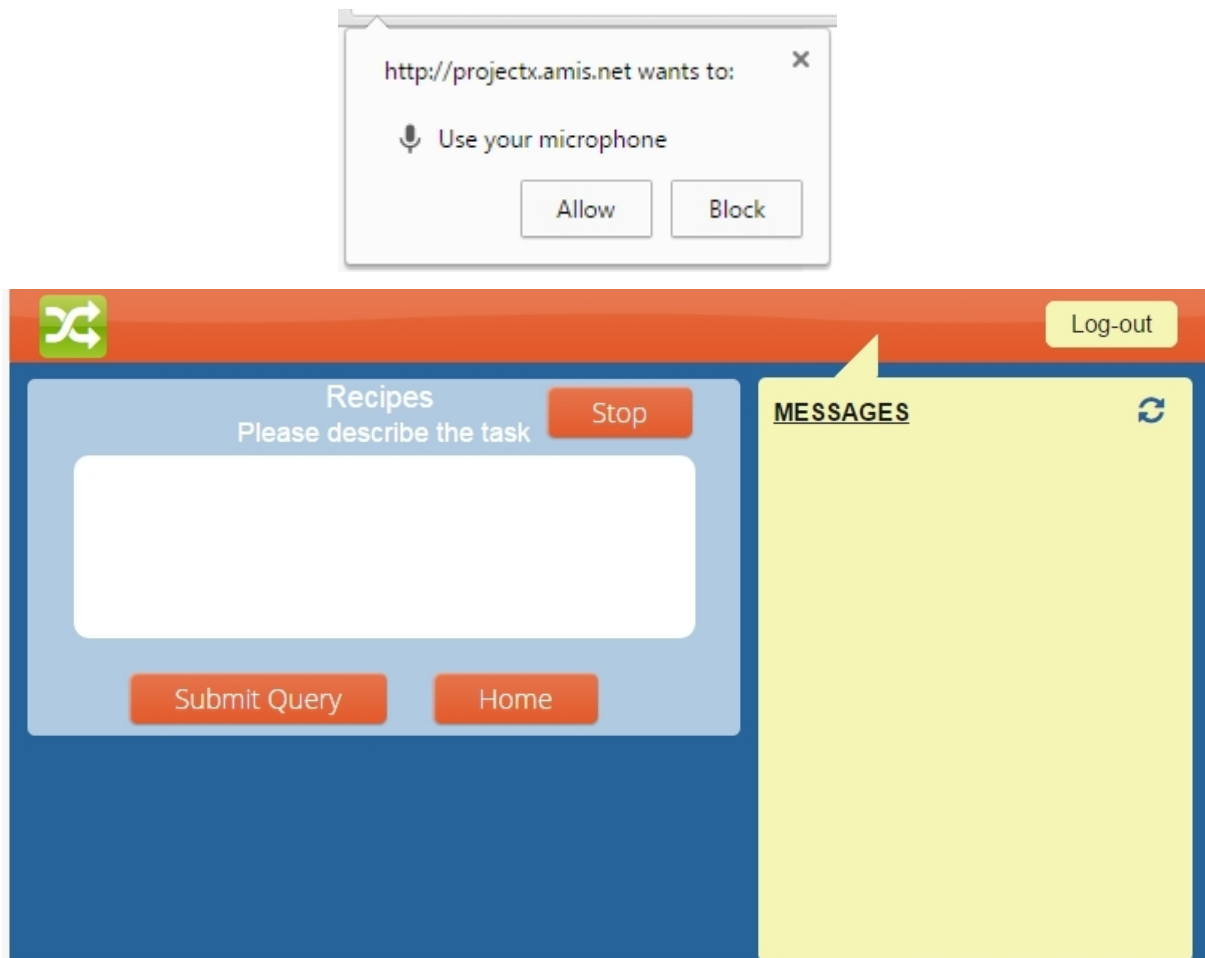


Figure 20 Browser natural language input

Alternatively by clicking the “Dictate” button and allowing his browser to use his microphone (Figure 20) he can dictate his task request to the SandS voice recognition engine.

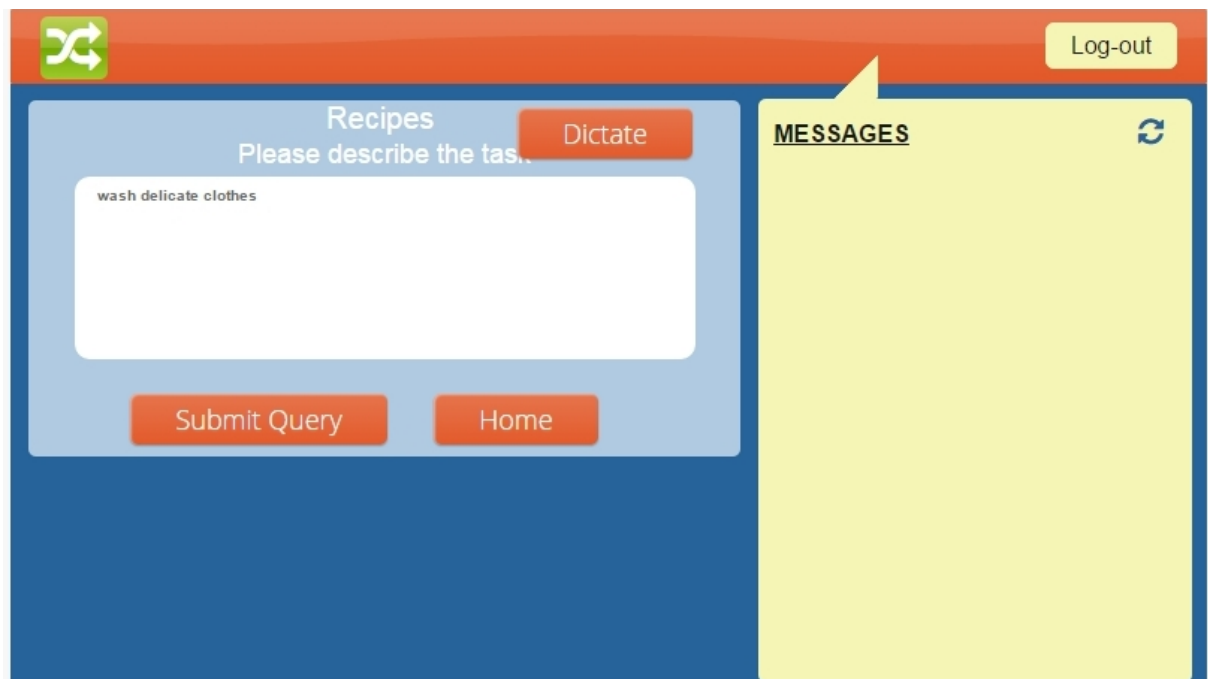
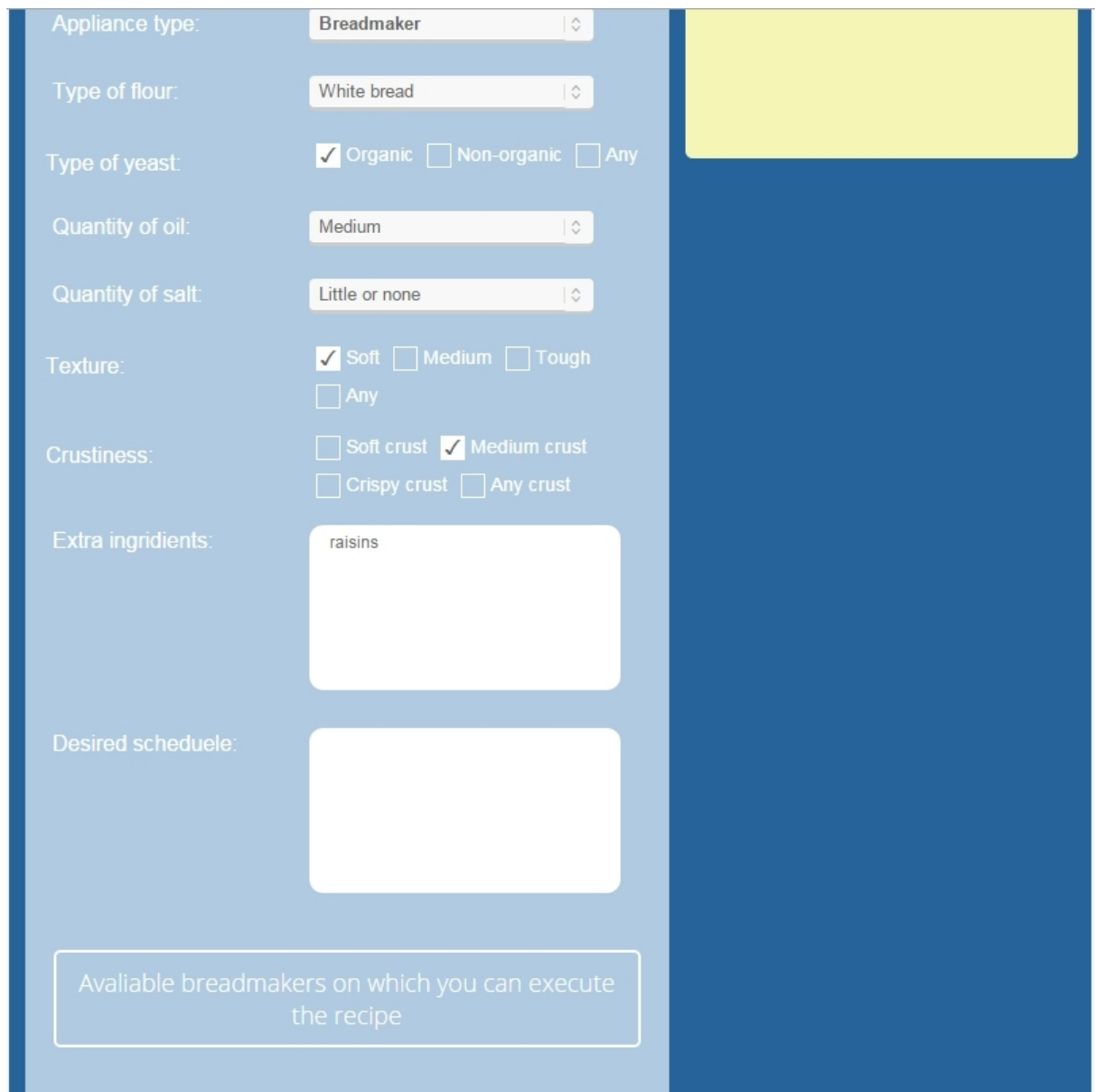


Figure 21 Task query confirmation and submission

Figure 21 depicts the result (“wash delicate clothes”) of a successfully recognized voice dictated task request. Hitting “Submit Query” will submit the query to the NI, which will return the best suited recipe for the specific user and task request.

3.2 Recipe approval/modification



The screenshot displays a recipe configuration interface with the following elements:

- Appliance type:** Breadmaker (dropdown menu)
- Type of flour:** White bread (dropdown menu)
- Type of yeast:** ☒ Organic ☐ Non-organic ☐ Any
- Quantity of oil:** Medium (dropdown menu)
- Quantity of salt:** Little or none (dropdown menu)
- Texture:** ☒ Soft ☐ Medium ☐ Tough ☐ Any
- Crustiness:** ☐ Soft crust ☒ Medium crust ☐ Crispy crust ☐ Any crust
- Extra ingredients:** raisins (text input field)
- Desired schedule:** (empty text input field)
- Bottom section:** A box containing the text "Available breadmakers on which you can execute the recipe".

Figure 22 Resulting recipe overview

Figure 22 illustrates the overview of a recipe returned by the NI for the breadmaker request “bake white bread with raisins”. Of course, the user can directly modify all recipe parameters such as Bread Texture, Crustiness, Extra ingredients and so on.

RECIPIE

WHITE_BREAD(complex)

Total: 02 hours 34 min 13 s

FIRSTLEAVENING time[31 min 08 s],
temperature[34.475673130693]

SECONBLEAVENING time[45 min 52 s],
temperature[35.106203706726]

PRECOOKING time[10 min 08 s],
temperature[61.461461795609]

COOKING time[01 hours 04 min 59 s],
temperature[118.87166225678]

BROWNING time[02 min 04 s],
temperature[130.04323714969]

Long-Term Scheduling

+

MESSAGES

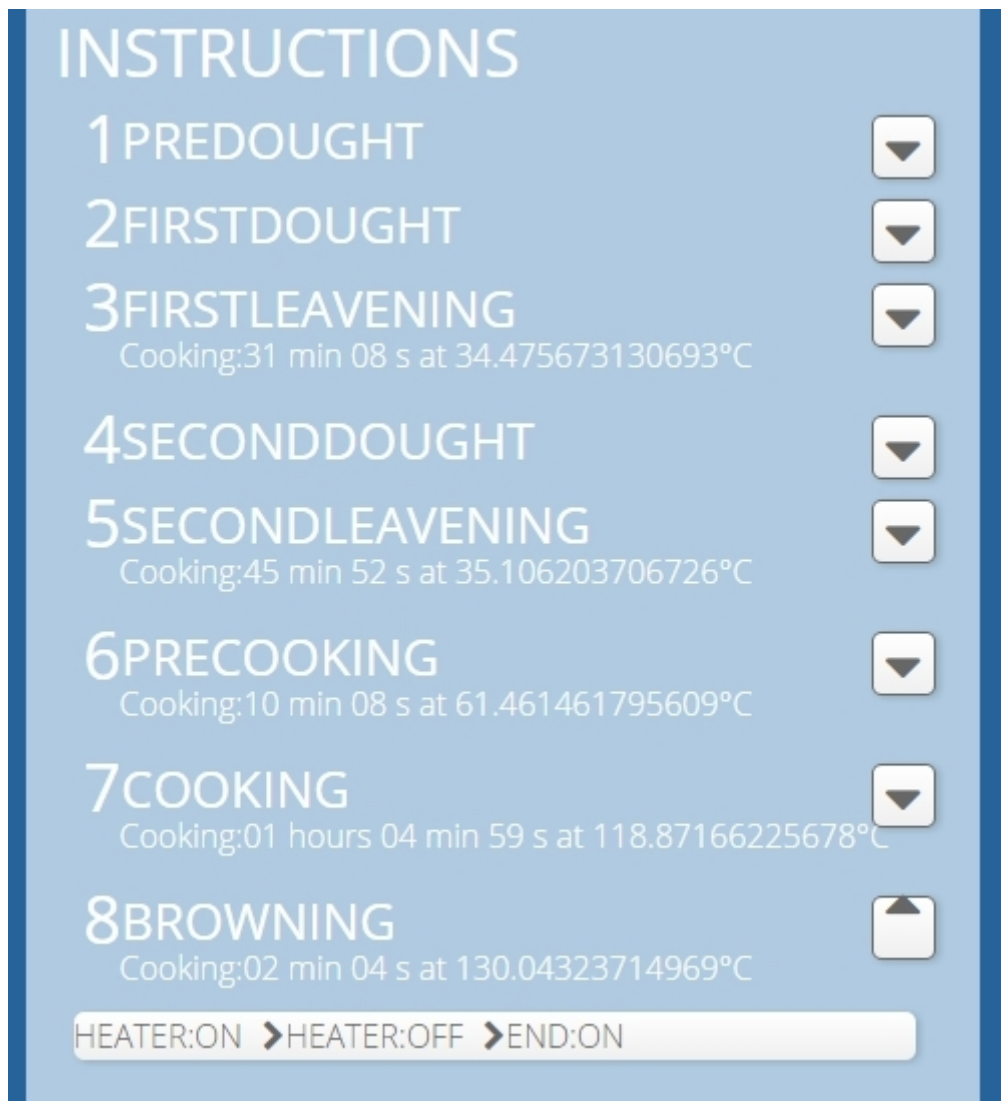


Figure 23 Detailed recipe phases with scheduling option

Figure 23 depicts the detailed recipe phases. The total time for the execution of the recipe is 2 hours and 35 minutes and the recipe consists of 8 different phases starting with “Predough” and ending with “Browning”. By clicking on a recipe phase, detailed information on the executed appliance instructions is available (“HEATER:ON”, “HEATER:OFF”). Please also note that the recipe can be directly added to the long term scheduler by clicking on the plus sign.

Refine parameters for BREADMAKER

In order to obtain a recipe more suitable to you, please re-adjust the parameters provided below

Brown bread ▼

Please select ▼

Please select

FLOUR

White bread

Please select

Brown bread

WATER

AMMOUNT

TEXTURE

CRUST

OIL

SALT

☐ Use extra parameters

Show extra parameters

Search

Cancel

Figure 24 Recipe refinement form

Furthermore, as depicted in Figure 24, the user can refine recipes according to his wishes using intuitive sliders, such as “Crust”, “Texture” etc.

3.3 Messaging



Figure 25 Recipe dispatch confirmation

After a recipe execution, SandS messaging system brings up a message in the Home Screen confirming that the recipe has been successfully terminated.

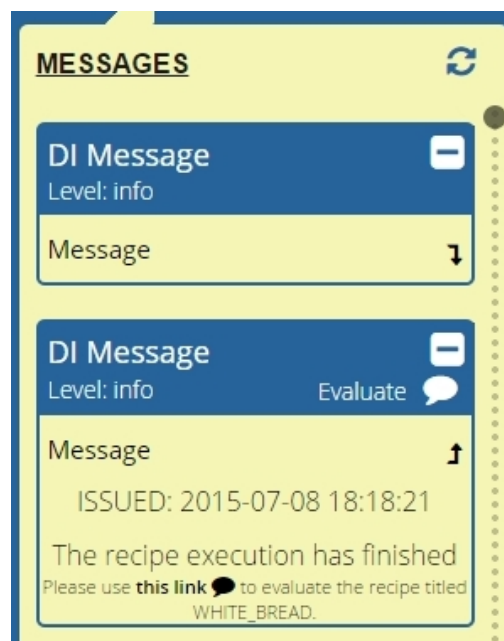


Figure 26 Detailed messaging board with evaluation link

A new DI message at the message board confirms that the recipe execution has finished. On the upper right corner of the message, an evaluation link is provided.

3.4 Evaluation

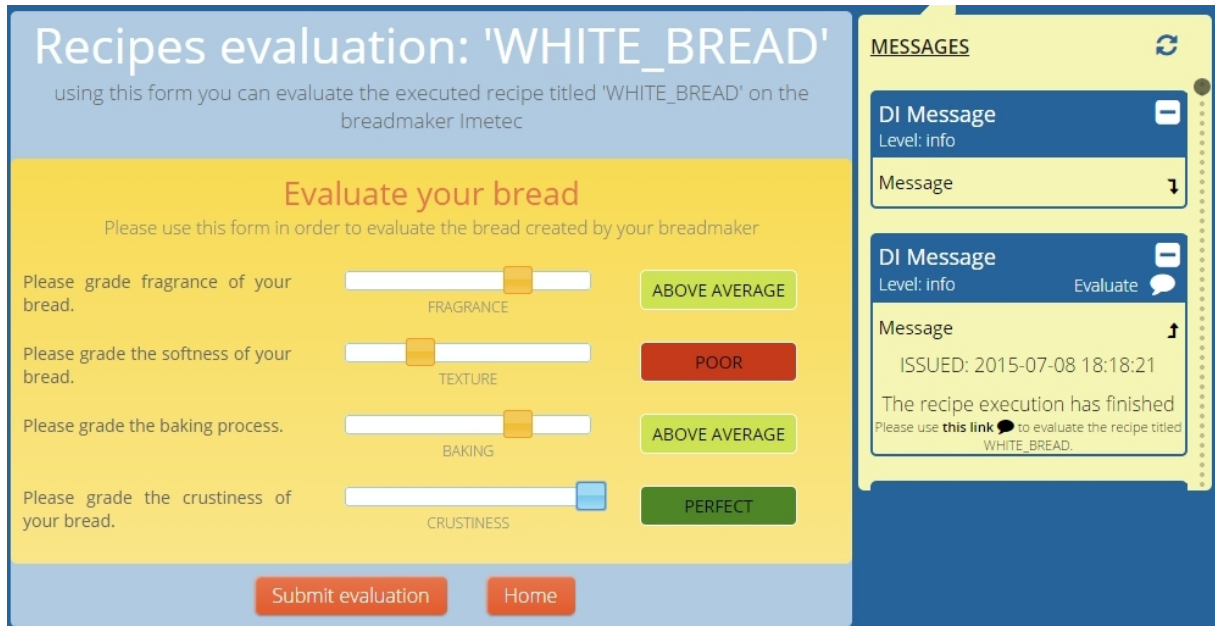


Figure 27 Recipe execution user feedback form

By clicking on the evaluation link of Figure 26 the user brings up the recipe execution feedback form of Figure 27. By moving sliders for various aspects of the recipe execution (“fragrance”, “softness”, “baking process”, “crustiness”) the user can provide feedback to the SandS NI.